Amendments to the Claims

Claim 1 (currently amended): A computer program product embodied on computer readable media readable by a computing system in a computing environment, for enforcing security policy using style sheet processing, comprising:

computer-readable program code means for obtaining an input document;

~~one or more stored policy enforcement objects, wherein each of said stored policy enforcement objects specifies a security policy to be associated with zero or more elements of said input document;~~

computer-readable program code means for obtaining a Document Type Definition (DTD) ~~corresponding to~~ that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD ~~has been augmented with one or more~~ references ~~to selected ones~~ one of a plurality of ~~said~~ stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element or elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

computer-readable program code means for applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

computer-readable program code means for creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables each community member authorized to view that element to use key distribution material associated with the output document when decrypting the encrypted element.

~~an augmented style sheet processor, wherein said augmented processor further comprises:~~

~~computer-readable program code means for loading said DTD;~~

~~computer-readable program code means for resolving each of said one or more references in said loaded DTD;~~

~~computer-readable program code means for instantiating said policy enforcement objects associated with said resolved references;~~

~~computer-readable program code means for executing selected ones of said instantiated policy enforcement objects during application of one or more style sheets to said input document, wherein a result of said computer-readable program code means for executing is an interim transient document reflecting said execution;~~

~~computer-readable program code means for generating one or more random encryption keys;~~

~~computer-readable program code means for encrypting selected elements of said interim transient document, wherein a particular one of said generated random encryption keys may be used to encrypt one or more of said selected elements, while leaving zero or more other elements of said interim transient document unencrypted;~~

~~computer-readable program code means for encrypting each of said one or more random encryption keys; and~~

~~computer-readable program code means for creating an encrypted output document comprising said zero or more other unencrypted elements, said selected encrypted elements, and said encrypted encryption keys;~~

~~computer-readable program code means for receiving said encrypted output document at a client device;~~

~~an augmented document processor, comprising computer-readable program code means for decrypting said received output document for an individual user or process on said client device, thereby creating a result document; and~~

~~computer-readable program code means for rendering said result document on said client device.~~

Claim 2 (currently amended): The computer program product according to Claim 1, wherein said markup notation in said interim transient document comprises ~~one or more encryption~~ tags ~~identifying elements needing encryption~~ of a markup language.

Claim 3 (original): The computer program product according to Claim 1, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 4 (original): The computer program product according to Claim 3, wherein said output document is specified in said XML notation.

Claim 5 (currently amended): The computer program product according to Claim 1, wherein said stored policy enforcement objects further comprise computer-readable program code means for overriding a method for evaluating said elements of said input document, and wherein said computer-readable program code means for applying said one or more style sheets ~~executing~~ further comprises computer-readable program code means for invoking ~~executing~~ said computer-readable program code means for overriding, thereby causing said markup notation to be added.

Claim 6 (original): The computer program product according to Claim 5, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 7 (original): The computer program product according to Claim 6, wherein said method is a value-of method of said XSL notation, and wherein said computer-readable program code means for overriding said value-of method is by subclassing said value-of method.

Claim 8 (currently amended): The computer program product according to Claim 5 ~~or Claim 7~~, wherein:

said ~~overridden~~ overriding method comprises:

computer-readable program code means for generating said markup notation as encryption tags; and

computer-readable program code means for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null ~~are determined to require~~ encryption requirement; and

said computer-readable program code means for creating said output document further comprises computer-readable program code means for encrypting ~~selected elements encrypts~~ those elements surrounded by said inserted encryption tags.

Claim 9 (canceled)

Claim 10 (currently amended): The computer program product according to ~~Claim 9, wherein~~ Claim 1, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 11 (currently amended): The computer program product according to ~~Claim 9, wherein~~ Claim 1, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 12 (currently amended): The computer program product according to ~~Claim 9, wherein~~ Claim 1, wherein said computer-readable program code means for creating said output document further comprises:

computer-readable program code means for generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

~~said~~ computer-readable program code means for encrypting said distinct symmetric ~~encryption~~ keys ~~further comprises computer-readable program code means for encrypting a different version of each of said random encryption keys~~ separately for each of ~~one or more~~ said members of ~~each of zero or more of~~ said ~~communities~~ community for which ~~uses~~ said ~~encryption~~ symmetric key was generated, thereby creating member-specific versions of each of said distinct symmetric keys. ~~, and wherein each of said different versions is encrypted using a public key of said community member for which said different version was encrypted.~~

Claim 13 (currently amended): The computer program product according to ~~Claim 9, wherein said encryption requirement may have a null value to indicate that said specified security policy does not require encryption.~~ Claim 12, wherein said computer-readable program code means for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions.

Claim 14 (currently amended): The computer program product according to Claim 1, wherein said ~~computer-readable program code means for encrypting selected~~ encrypted elements in said created output document are encrypted using ~~uses~~ a cipher block chaining mode encryption process.

Claim 15 (currently amended): The computer program product according to Claim 12, further comprising:

computer-readable program code means for creating a key class for each of said unique ~~community~~ communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community ~~is an~~ are authorized ~~viewer~~ viewers, and wherein said key class comprises: (1) ~~a strongest~~ an encryption algorithm identifier and key length used when encrypting ~~requirement of~~ said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said ~~different~~ member-specific versions of said encrypted symmetric ~~encryption~~ key for each of said identified community members~~; and~~

~~wherein:~~

~~said computer-readable program code means for generating said one or more random encryption keys generates a particular one of said random encryption keys for each of said key classes, and wherein each of said different versions in a particular key class is encrypted from said generated encryption key generated for said key class; and~~

~~said computer-readable program code means for encrypting selected elements uses that one of said particular random encryption keys which was generated for said key class with which said selected element is associated~~.

Claim 16 (currently amended): The computer program product according to Claim 12, further comprising ~~wherein~~:

~~said~~ computer-readable program code means for decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further ~~comprises~~ comprising:

computer-readable program code means for determining zero or more of said communities of which said individual user or process is one of said members;

computer-readable program code means for decrypting, for each of said determined communities, said ~~different~~ member-specific version of said ~~random encrytion~~ symmetric key ~~which was encrypted using said public key of said one member, wherein said computer-readable program code means for decrypting uses a private key of said one member which is associated with said public key which was used for encryption~~, thereby creating a decrypted key; and

computer-readable program code means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities~~; and~~

~~said computer-readable program code means for rendering further comprises:~~

~~computer-readable program code means for rendering said decrypted selected ones and said other unencrypted elements~~.

Claim 17 (currently amended): The computer program product according to Claim 15, wherein said computer-readable program code means for encrypting each of said distinct symmetric keys

separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions and further comprising:

~~said~~ computer-readable program code means for decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further ~~comprises~~ comprising:

computer-readable program code means for determining zero or more of said key classes which identify said individual user or process as one of said members;

computer-readable program code means for decrypting, for each of said determined key classes, said ~~different~~ member-specific version of said ~~random encrytion~~ encrypted symmetric key, using ~~key in said key class which was encrypted using said public key of said one member, wherein said computer-readable program code means for decrypting uses~~ a private key of said individual user or process, ~~one member which is associated with said public key which was used for encryption,~~ thereby creating a decrypted key; and

computer-readable program code means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined key classes. ~~class; and~~

~~said computer-readable program code means for rendering further comprises:~~

~~computer-readable program code means for rendering said decrypted selected ones and said other unencrypted elements.~~

Claim 18 (currently amended): The computer program product according to Claim 16 or Claim 17, ~~wherein said computer-readable program code means for rendering further comprises~~ further comprising computer-readable program code means for substituting a predetermined ~~rendering a substitute~~ text message for any ~~of said selected~~ encrypted elements in said output document which cannot be decrypted ~~by said computer-readable program code means for decrypting said output document~~ for said individual user or process.

Claim 19 (original): The computer program product according to Claim 1, wherein said DTD is replaced by a schema.

Claim 20 (currently amended); The computer program product according to ~~Claim 9, wherein~~ Claim 1, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 21 (original): The computer program product according to Claim 8, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

Claim 22 (currently amended): A system for enforcing security policy using style sheet processing in a computing environment, comprising:

an input document;

~~one or more stored policy enforcement objects, wherein each of said stored policy enforcement objects specifies a security policy to be associated with zero or more elements of said input document;~~

a Document Type Definition (DTD) ~~corresponding to~~ that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD ~~has been augmented with one or more~~ references ~~to selected ones~~ one of a plurality of ~~said~~ stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element of elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

means for applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

means for creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables each community member authorized to view that element to use key distribution material associated with the output document when decrypting the encrypted element.

~~an augmented style sheet processor, wherein said augmented processor further comprises:~~

~~means for loading said DTD;~~

~~means for resolving each of said one or more references in said loaded DTD;~~

~~means for instantiating said policy enforcement objects associated with said resolved references;~~

~~means for executing selected ones of said instantiated policy enforcement objects during application of one or more style sheets to said input document, wherein a result of said means for executing is an interim transient document reflecting said execution;~~

~~means for generating one or more random encryption keys;~~

~~means for encrypting selected elements of said interim transient document, wherein a particular one of said generated random encryption keys may be used to encrypt one or more of said selected elements, while leaving zero or more other elements of said interim transient document unencrypted;~~

~~means for encrypting each of said one or more random encryption keys; and~~

~~means for creating an encrypted output document comprising said zero or more other unencrypted elements, said selected encrypted elements, and said encrypted encryption keys;~~

~~means for receiving said encrypted output document at a client device;~~

~~an augmented document processor, comprising means for decrypting said received output document for an individual user or process on said client device, thereby creating a result document; and~~

~~means for rendering said result document on said client device.~~

Claim 23 (currently amended): The system according to Claim 22, wherein said markup notation in said interim transient document comprises ~~one or more encryption~~ tags ~~identifying elements needing encryption~~ of a markup language.

Claim 24 (original): The system according to Claim 22, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 25 (original): The system according to Claim 24, wherein said output document is specified in said XML notation.

Claim 26 (currently amended): The system according to Claim 22, wherein said stored policy enforcement objects further comprise means for overriding a method for evaluating said elements of said input document, and wherein said means for ~~executing~~ applying said one or more style sheets further comprises means for ~~executing~~ invoking said means for overriding, thereby causing said markup notation to be added.

Claim 27 (original): The system according to Claim 26, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 28 (original): The system according to Claim 27, wherein said method is a value-of method of said XSL notation, and wherein said means for overriding said value-of method is by subclassing said value-of method.

Claim 29 (currently amended): The system according to Claim 26 ~~or Claim 28~~, wherein:

said ~~overridden~~ overriding method comprises:

means for generating said markup notation as encryption tags; and

means for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null ~~are determined to require~~ encryption requirement; and

said means for creating said output document further comprises means for encrypting ~~selected elements encrypts~~ those elements surrounded by said inserted encryption tags.

Claim 30 (canceled)

Claim 31 (currently amended): The system according to ~~Claim 30, wherein~~ Claim 22, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 32 (currently amended): The system according to ~~Claim 30, wherein~~ Claim 22, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 33 (currently amended): The system according to ~~Claim 30, wherein~~ Claim 22, wherein said means for creating said output document further comprises:

means for generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

~~said~~ means for encrypting said distinct symmetric ~~encryption~~ keys ~~further comprises means for encrypting a different version of each of said random encryption keys~~ separately for each of ~~one or more~~ said members of ~~each of zero or more of~~ said ~~communities~~ community for which ~~uses~~ said ~~encryption~~ symmetric key was generated, thereby creating member-specific versions of each of said distinct symmetric keys. ~~, and wherein each of said different versions is encrypted using a public key of said community member for which said different version was encrypted.~~

Claim 34 (currently amended): The system according to ~~Claim 30, wherein said encryption requirement may have a null value to indicate that said specified security policy does not require encryption.~~ Claim 33, wherein said means for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions.

Claim 35 (currently amended): The system according to Claim 22, wherein said ~~means for encrypting selected~~ encrypted elements in said created output document are encrypted using ~~uses~~ a cipher block chaining mode encryption process.

Claim 36 (currently amended): The system according to Claim 33, further comprising:

means for creating a key class for each of said unique ~~community~~ communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community ~~is an~~ are authorized ~~viewer~~ viewers, and wherein said key class comprises: (1) ~~a strongest~~ an encryption algorithm identifier and key length used when encrypting ~~requirement of~~ said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said ~~different~~ member-specific versions of said encrypted symmetric ~~encryption~~ key for each of said identified community members~~; and~~

~~wherein:~~

~~said means for generating said one or more random encryption keys generates a particular one of said random encryption keys for each of said key classes, and wherein each of said different versions in a particular key class is encrypted from said generated encryption key generated for said key class; and~~

~~said means for encrypting selected elements uses that one of said particular random encryption keys which was generated for said key class with which said selected element is associated~~.

Claim 37 (currently amended): The system according to Claim 33, further comprising ~~wherein~~:

~~said~~ means for decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further ~~comprises~~ comprising:

means for determining zero or more of said communities of which said individual user or process is one of said members;

means for decrypting, for each of said determined communities, said member-specific ~~different~~ version of said ~~random encrytion~~ symmetric key ~~which was encrypted using said public key of said one member, wherein said means for decrypting uses a private key of said one member which is associated with said public key which was used for encryption~~, thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities~~; and~~

~~said means for rendering further comprises:~~

~~means for rendering said decrypted selected ones and said other unencrypted elements~~.

Claim 38 (currently amended): The system according to Claim 36, wherein said means for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions and further comprising:

~~said~~ means for decrypting, for an individual user of process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further ~~comprises~~ comprising:

means for determining zero or more of said key classes which identify said individual user or process as one of said members;

means for decrypting, for each of said determined key classes, said member-specific ~~different~~ version of said encrypted symmetric key, using ~~random encrytion key in said key class which was encrypted using said public key of said one member, wherein said means for decrypting uses~~ a private key of said individual user or process, ~~one member which is associated with said public key which was used for encryption,~~ thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined key classes. ~~class; and~~

~~said means for rendering further comprises:~~

~~means for rendering said decrypted selected ones and said other unencrypted elements.~~

Claim 39 (currently amended): The system according to Claim 37 or Claim 38, ~~wherein said means for rendering further comprises~~ further comprising means for substituting a predetermined ~~rendering a substitute~~ text message for ~~any of said selected~~ encrypted elements in said output document which cannot be decrypted ~~by said means for decrypting said output document~~ for said individual user or process.

Claim 40 (original): The system according to Claim 22, wherein said DTD is replaced by a schema.

Claim 41 (currently amended): The system according to ~~Claim 30, wherein~~ Claim 22, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 42 (original): The system according to Claim 29, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

Claim 43 (currently amended): A method for enforcing security policy using style sheet processing in a computing environment, comprising the steps of:

providing an input document;

~~providing one or more stored policy enforcement objects, wherein each of said stored policy enforcement objects specifies a security policy to be associated with zero or more elements of said input document;~~

providing a Document Type Definition (DTD) ~~corresponding to~~ that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD ~~has been augmented with one or more~~ references ~~to selected ones~~ one of a plurality of ~~said~~ stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element of elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables each community member authorized to view that element to use key distribution material associated with the output document when decrypting the encrypted element.

~~executing an augmented style sheet processor, further comprising the steps of:~~

~~loading said DTD;~~

~~resolving each of said one or more references in said loaded DTD;~~

~~instantiating said policy enforcement objects associated with said resolved references;~~

~~executing selected ones of said instantiated policy enforcement objects during application of one or more style sheets to said input document, wherein a result of said executing selected ones step is an interim transient document reflecting said execution;~~

~~generating one or more random encryption keys;~~

~~encrypting selected elements of said interim transient document, wherein a particular one of said generated random encryption keys may be used to encrypt one or more of said selected elements, while leaving zero or more other elements of said interim transient document unencrypted;~~

~~encrypting each of said one or more random encryption keys; and~~

~~creating an encrypted output document comprising said zero or more other unencrypted elements, said selected encrypted elements, and said encrypted encryption keys;~~

~~receiving said encrypted output document at a client device;~~

~~executing an augmented document processor, comprising the step of decrypting said received output document for an individual user or process on said client device, thereby creating a result document; and~~

~~rendering said result document on said client device.~~

Claim 44 (currently amended): The method according to Claim 43, wherein said markup notation in said interim transient document comprises ~~one or more encryption~~ tags ~~identifying elements needing encryption~~ of a markup language.

Claim 45 (original): The method according to Claim 43, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 46 (original): The method according to Claim 45, wherein said output document is specified in said XML notation.

Claim 47 (currently amended): The method according to Claim 43, wherein said stored policy enforcement objects further comprise executable code for overriding a method for evaluating said elements of said input document, and wherein said ~~executing selected ones~~ applying step further

comprises overriding said method for evaluating, thereby causing said markup notation to be added.

Claim 48 (original): The method according to Claim 47, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 49 (original): The method according to Claim 48, wherein said method is a value-of method of said XSL notation, and wherein said step of overriding said value-of method is by subclassing said value-of method.

Claim 50 (currently amended): The method according to Claim 47 ~~or Claim 49~~, wherein:

said step of overriding further comprises the steps of:

generating said markup notation as encryption tags; and

inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null ~~are determined to require~~ encryption requirement; and

said step of creating said output document further comprises the step of encrypting ~~selected elements encrypts~~ those elements surrounded by said inserted encryption tags.

Claim 51 (canceled)

Claim 52 (currently amended): The method according to ~~Claim 51, wherein~~ Claim 43, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that policy.

Claim 53 (currently amended): The method according to ~~Claim 51, wherein~~ Claim 43, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that policy.

Claim 54 (currently amended): The method according to ~~Claim 51, wherein~~ Claim 43, wherein said step of creating said output document further comprises the steps of:

generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

~~said step of~~ encrypting said distinct symmetric ~~encryption~~ keys ~~further comprises the step of encrypting a different version of each of said random encryption keys~~ separately for each of ~~one or more~~ said members of ~~each of zero or more of~~ said ~~communities~~ community for which ~~uses~~ said ~~encryption~~ symmetric key was generated, thereby creating member-specific versions of each of said distinct symmetric keys. ~~, and wherein each of said different versions is encrypted using a public key of said community member for which said different version was encrypted.~~

Claim 55 (currently amended): The method according to ~~Claim 51, wherein said encryption requirement may have a null value to indicate that said specified security policy does not require~~

~~encryption.~~ Claim 54, wherein said step of encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions.

Claim 56 (currently amended): The method according to Claim 43, wherein said ~~step of encrypting selected~~ encrypted elements in said created output document are encrypting using ~~uses~~ a cipher block chaining mode encryption process.

Claim 57 (currently amended): The method according to Claim 54, further comprising the step of:

creating a key class for each of said unique ~~community~~ communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community ~~is an~~ are authorized ~~viewer~~ viewers, and wherein said key class comprises: (1) ~~a strongest~~ an encryption algorithm identifier and key length used when encrypting ~~requirement of~~ said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said ~~different~~ member-specific versions of said encrypted symmetric ~~encryption~~ key for each of said identified community members~~; and~~

~~wherein:~~

~~said step of generating said one or more random encryption keys generates a particular one of said random encryption keys for each of said key classes, and wherein each of said different versions in a particular key class is encrypted from said generated encryption key generated for said key class; and~~

~~said step of encrypting selected elements uses that one of said particular random encryption keys which was generated for said key class with which said selected element is associated~~.

Claim 58 (currently amended): The method according to Claim 54, further comprising the step of ~~wherein~~:

~~said step of~~ decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further ~~comprises~~ comprising the steps of:

determining zero or more of said communities of which said individual user or process is one of said members;

decrypting, for each of said determined communities, said ~~different~~ member-specific version of said ~~random encrytion~~ symmetric key ~~which was encrypted using said public key of said one member, wherein said step of decrypting uses a private key of said one member which is associated with said public key which was used for encryption~~, thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities~~; and~~

~~said step of rendering further comprises the step of:~~

~~rendering said decrypted selected ones and said other unencrypted elements~~.

59. The method according to Claim 57, wherein said step of encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions and further comprising the step of:

~~said step of~~ decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further ~~comprises~~ comprising the steps of:

determining zero or more of said key classes which identify said individual user or process as one of said members;

decrypting, for each of said determined key classes, said ~~different~~ member-specific version of said ~~random encrytion~~ encrypted symmetric key, using ~~key in said key class which was encrypted using said public key of said one member, wherein said step of decrypting uses~~ a private key of said individual user or process, ~~one member which is associated with said public key which was used for encryption,~~ thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined key classes. ~~; and~~

~~said step of rendering further comprises the step of:~~

~~rendering said decrypted selected ones and said other unencrypted elements.~~

Claim 60 (currently amended): The method according to Claim 58 or Claim 59, ~~wherein said step of rendering further comprises~~ further comprising the step of substituting a predetermined ~~rendering a substitute~~ text message for any ~~of said selected~~ encrypted elements in said output

document which cannot be decrypted for said individual user or process ~~by said step of decrypting said output document~~.

Claim 61 (original): The method according to Claim 43, wherein said DTD is replaced by a schema.

Claim 62 (currently amended): The method according to ~~Claim 51, wherein~~ Claim 43, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 63 (original): The method according to Claim 50, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.